

| Apr 28, 00 12:47   | Recursive.m3 | Page 1/1 |
|--|--------------|----------|
| <pre>MODULE Recursive EXPORTS Main ; IMPORT IO ;  5  PROCEDURE BinKoeff(n, m : INTEGER) : INTEGER =     BEGIN       IF m = 0 THEN         RETURN 1 ;       ELSEIF n = m THEN         RETURN 1 ;       ELSE         RETURN BinKoeff(n-1, m-1) + BinKoeff(n-1, m) ;       END ;     END BinKoeff ;  15  CONST N = 10 ;  20  BEGIN     FOR n := 0 TO N DO       FOR m := 0 TO n DO         IO.PutInt(BinKoeff(n, m)) ;         IO.Put(" ") ;       END ;       IO.Put("\n") ;     END ;     END Recursive .</pre> |              |          |

| May 03, 00 18:15   | NonRecursive.m3 | Page |
|--|-----------------|------|
| <pre>MODULE NonRecursive EXPORTS Main ; (* Variante 1: quadratische Speicherplatzkomplexitaet *) 5  IMPORT IO ;  CONST N = 10 ;  10  VAR a : ARRAY [0..N] OF ARRAY [0..N] OF INTEGER ;     BEGIN       FOR n := 0 TO 10 DO         FOR m := 0 TO n DO           IF m = 0 THEN             a[n][m] := 1 ;           ELSEIF n = m THEN             a[n][m] := 1 ;           ELSE             a[n][m] := a[n-1][m-1] + a[n-1][m] ;           END ;           IO.PutInt(a[n][m]) ;           IO.Put(" ") ;         END ;         IO.Put("\n") ;       END ;       END NonRecursive .</pre> |                 |      |

| May 03, 00 18:15   | NonRecursive2.m3 | Page 1/1 |
|--|------------------|----------|
| <pre>MODULE NonRecursive2 EXPORTS Main ; (* Variante 2: weniger gut lesbar, dafuer bessere  * Speicherplatzausnutzung O(n) *) 5  IMPORT IO ;  CONST N = 10 ;  10  VAR a : ARRAY [0..N] OF INTEGER ;     BEGIN       FOR n := 0 TO 10 DO         FOR m := n TO 0 BY -1 DO           IF m = 0 THEN             a[m] := 1 ;           ELSEIF n = m THEN             a[m] := 1 ;           ELSE             a[m] := a[m-1] + a[m] ;           END ;           IO.PutInt(a[m]) ;           IO.Put(" ") ;         END ;         IO.Put("\n") ;       END ;       END NonRecursive2 .</pre> |                  |          |

| May 03, 00 18:08  | output.txt | Page |
|---|------------|------|
| <pre>1 1 1 1 2 1 1 1 3 3 1 5 1 4 6 4 1 1 5 10 10 5 1 1 6 15 20 15 6 1 1 7 21 35 35 21 7 1 10 1 8 28 56 70 56 28 8 1 1 9 36 84 126 126 84 36 9 1 1 10 45 120 210 252 210 120 45 10 1</pre> |            |      |